



РОЗРОБКА СЕРВЕРНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	4 курс, осінній семестр
Обсяг дисципліни	4 кредити, 120 годин, з яких 72 години аудиторних (36 год. лекцій), 18 год. практичні
Семестровий контроль/ контрольні заходи	Залік.
Розклад занять	http://rozklad.kpi.ua/ 1 лекція (2 години) 1 раз на тиждень; 1 практичне заняття (2 години) 1 раз на 2 тижні.
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д.т.н., проф. Барабаш Олег Володимирович, bar64@ukr.net Практичні заняття: асистент Макарчук Андрій Валентинович, makarchukandriy1999@gmail.com
Розміщення курсу	Кампус

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна «Розробка серверного програмного забезпечення» є тією частиною знань, що формує фахівця з інженерії програмного забезпечення.

Метою навчальної дисципліни є формування у студентів знань, навичок та компетенцій з розробки серверного програмного забезпечення за допомогою мови програмування Java.

Предметом вивчення є основи розробки серверного програмного забезпечення за допомогою мови програмування Java, а також прикладні аспекти серверного програмного забезпечення.

Програмні результати

Результатом вивчення навчальної дисципліни є формування у студентів **компетентностей**: — здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);

— здатність застосовувати знання у практичних ситуаціях (ЗК 02);

— здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 06);

— володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 07);

— здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 08);

— здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення (ФК 12);

— здатність до алгоритмічного та логічного мислення (ФК 14).

Згідно з вимогами програми навчальної дисципліни студенти після вивчення дисципліни мають продемонструвати такі результати навчання:

— аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);

— вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання (ПРН 11);

— знати і застосовувати методи розробки алгоритмів, конструювання серверного програмного забезпечення за допомогою мови програмування Java (ПРН 13).

— вміти організовувати, налаштовувати та програмувати у комп'ютерних мережах (ПРН 27).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Згідно з робочим навчальним планом навчальна дисципліна «Розробка серверного програмного забезпечення» викладається студентам четвертого року підготовки першого (бакалаврського) рівня вищої освіти ступеня “бакалавр” спеціальності “Інженерія програмного забезпечення” освітньої програми “Інженерія програмного забезпечення інтелектуальних кіберфізичних систем в енергетиці” у першому навчальному семестрі. Дисципліна «Розробка серверного програмного забезпечення» базується на знаннях дисциплін «Основи програмування», «Алгоритми та структури даних», «Компоненти програмної інженерії», «Бази даних», «Об'єктно-орієнтований аналіз та конструювання програмних систем», «Мережеві технології», «Архітектура системного програмного забезпечення». Вона забезпечує вивчення таких навчальних дисциплін як: «Побудова масштабованих систем обробки даних у реальному часі», «Переддипломна практика», «Дипломне проектування».

3. Зміст навчальної дисципліни

Розділ 1. Основні поняття серверного програмного забезпечення

Тема 1. Вступ до серверного програмного забезпечення

Тема 2. Основи Java для серверної розробки

Тема 3. Шаблони проєктування для серверних додатків

Тема 4. Моделі обробки запитів на стороні сервера: синхронна та асинхронна обробка

Тема 5. Основи створення веб-сервера (без фреймворків)

Розділ 2. Основи розробки серверного програмного забезпечення на Java.

Тема 6. Основи фреймворку Spring

Тема 7. Розробка RESTful API з використанням Spring

Тема 8. Підключення до баз даних через Spring Data JPA

Тема 9. Основи безпеки серверних додатків

Тема 10. Spring Security: концепції та реалізація

Тема 11. Тестування серверних додатків

Тема 12. Обмеження продуктивності: теоретичні та практичні аспекти

Розділ 3. Прикладні аспекти розробки серверного програмування.

Тема 13. Кешування в серверних додатках

Тема 14. Основи мікросервісної архітектури

Тема 15. Взаємодія між мікросервісами

Тема 16. Розгортання серверного додатку та CI/CD.

Тема 17. Моніторинг та логування серверних додатків

Тема 18. Практичні аспекти масштабування серверних додатків

4. Навчальні матеріали та ресурси

Основна література:

1. Bloch J. Effective Java. 3rd ed. Boston: Addison-Wesley, 2023. 416 p.
2. Goetz B., Peierls T., Lea D., Bloch J., Bowbeer K., Holmes D. Java Concurrency in Boston: Addison-Wesley, 2024. 384 p.
3. Walls Craig Spring in Action. 5th ed. Shelter Island: Manning Publications, 2022. 520 p.
4. Sharma Sourabh. Mastering Microservices with Java. 2nd ed. Birmingham: Packt Publishing, 2022. 480 p.
5. Marchioni Francesco. Pro Java EE 8: An In-Depth Guide to Modern Web and Enterprise Development. New York: Apress, 2023. 661 p.
6. Rajesh R. V. Building RESTful Web Services with Spring 5. Birmingham: Packt Publishing, 2023. 490 p.
7. Oaks Scott. Java Performance: The Definitive Guide. 2nd ed. Sebastopol: O'Reilly Media, 2022. 456 p.

Додаткова література:

1. Gamma E., Helm R., Johnson R., Vlissides J., Gamma E. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 2024. 395 p.

2. Fowler Martin. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002. 533 p.
3. Spilca Laurentiu. Spring Security in Action. Shelter Island: Manning Publications, 2023. 576 p.
4. Hoffman Andrew. Web Application Security: Exploitation and Countermeasures for Modern Web Applications. Sebastopol: O'Reilly Media, 2023. 322 p.
5. Kim Gene, Behr Kevin, Spafford George. The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. 5th ed. Portland: IT Revolution Press, 2022. 432 p.
6. Humble Jez, Farley David. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment. Boston: Addison-Wesley, 2022. 512 p.
7. Long Josh, Bastani Kenny and Foundry Cloud. Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud. Sebastopol: O'Reilly Media, 2024. 648 p.
8. Newman Sam. Building Microservices: Designing Fine-Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2022. 615 p.
9. Keith Mike, Schincariol Merrick. Pro JPA 2 in Java EE 8: An In-Depth Guide to Java Persistence APIs. New York: Apress, 2024. 822 p.
10. Botros Silvia, Schwartz Baron, Zaitsev Peter. High-Performance MySQL, 4th ed. Sebastopol: O'Reilly Media, 2024. 852 p.
11. Beck Kent. Test-Driven Development: By Example. Boston: Addison-Wesley, 2022. 240 p.
12. Koskela Lasse. Unit Testing in Java: How Tests Drive the Code. Shelter Island: Manning Publications, 2023. 362 p.
13. Shirazi Jack. Java Performance Tuning. 2nd ed. Sebastopol: O'Reilly Media, 2023. 752 p.
14. Abbott Martin L., Fisher Michael T. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. 2nd ed. Upper Saddle River: Addison-Wesley, 2022. 624 p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Основні поняття серверного програмного забезпечення	
Тема 1. Вступ до серверного програмного забезпечення.	
1	Клієнт-серверна архітектура. Поняття та роль серверного ПЗ. Мови програмування та фреймворки, що використовуються для розробки серверного ПЗ.
Тема 2. Основи Java для серверної розробки.	
2	Огляд класів та API у Java, важливих для розробки серверних додатків. Версіювання та документування API.
Тема 3. Шаблони проєктування для серверних додатків	
3	Шаблони проєктування: Singleton, Factory, Builder, Observer. Їх застосування в серверній розробці. Приклади реалізації.
Тема 4. Моделі обробки запитів на стороні сервера: синхронна та асинхронна обробка.	
4	Різниця між синхронною та асинхронною моделлю обробки запитів. Блокуюча обробка запитів. Переваги неблокуючої модель при навантаженні. Поняття транзакцій.

Тема 5. Основи створення веб-сервера (без фреймворків).	
5	Використання ServerSocket у Java. Читання запиту, обробка рядків серверним веб-сервером. Побудова простої відповіді з HTML та JSON. Обмеження такого підходу – перехід до фреймворків.
Розділ 2. Основи розробки серверного програмного забезпечення на Java	
Тема 6. Основи фреймворку Spring.	
6	IoC (Inversion of Control) та Dependency Injection у Spring. Конфігурація додатків: Java-based, XML, Annotations. Вступ до Spring Boot.
Тема 7. Розробка RESTful API з використанням Spring.	
7	Основи RESTful сервісів. Створення контролерів та ресурсів у Spring. Методи HTTP: GET, POST, PUT, DELETE.
Тема 8. Підключення до баз даних через Spring Data JPA.	
8	Огляд Java Persistence API (JPA). Spring Data JPA: базові поняття. Створення репозиторіїв та робота з базами даних. Поняття ORM. Інші способи взаємодії з базами даних.
Тема 9. Основи безпеки серверних додатків.	
9	Основні принципи безпеки. Аутентифікація та авторизація. Загрози для серверних додатків.
Тема 10. Spring Security: концепції та реалізація.	
10	Використання Spring Security для захисту REST API. Огляд OAuth2 та JWT. Приклади конфігурації безпеки.
Тема 11. Тестування серверних додатків	
11	Основи юніт-тестування в Java. Тестування REST API з використанням MockMVC. Інтеграційні тести для Spring-додатків.
Тема 12. Обмеження продуктивності: теоретичні та практичні аспекти.	
12	Типові обмеження продуктивності серверних додатків. Потокова модель Java-серверів. Проблеми з пам'яттю і GC (Garbage Collection). Архітектурні рішення як відповідь на обмеження.
Розділ 3. Прикладні аспекти розробки серверного програмування	
Тема 13. Кешування в серверних додатках.	
13	Поняття кешування. Використання Spring Cache. Механізми кешування в розподілених системах.
Тема 14. Основи мікросервісної архітектури.	
14	Переваги та виклики мікросервісів. Принципи побудови мікросервісів на Java. Створення простих мікросервісів за допомогою Spring Boot.
Тема 15. Взаємодія між мікросервісами.	
15	Комунікація між мікросервісами через REST. Використання Spring Cloud для мікросервісної архітектури. API Gateway та Service Discovery.

Тема 16. Розгортання серверного додатку та CI/CD.	
16	Основи DevOps для серверного ПЗ. Налаштування CI/CD з Jenkins та GitLab. Контейнеризація додатків за допомогою Docker.
Тема 17. Моніторинг та логування серверних додатків.	
17	Інструменти для моніторингу: Prometheus, Grafana. Організація логування в Spring додатках. Аналіз логів і виявлення проблем.
Тема 18. Практичні аспекти масштабування серверних додатків.	
18	Принципи масштабування серверних додатків. Горизонтальне та вертикальне масштабування. Використання хмарних платформ для масштабування.

Практичні заняття

1	Оглядова розробка консольного Java-сервісу, що емулює серверні принципи Написання простої Java-програми, яка приймає "запит" (від користувача) та повертає "відповідь"; Розбір структури класу Main, базових методів вводу/виводу; Пояснення ролі API, структури клієнт-серверного взаємозв'язку.
2	Реалізація базових шаблонів (Factory, Singleton) та симуляція синхронної/асинхронної обробки запитів Реалізація Singleton, Factory та приклади їхнього використання в "серверному" коді; Створення 2 версій класу обробки запитів – синхронної та з імітацією асинхронності (Thread.sleep, ExecutorService); Аналіз поведінки при одночасному запуску декількох потоків.
3	Порівняння "ручного" сервера на ServerSocket та першого проєкту Spring Boot Написання простого серверного застосунку на ServerSocket для обробки текстових запитів; Створення "Hello World" сервісу у Spring Boot; Порівняння архітектури, складності, переваг і недоліків.
4	Розробка CRUD REST API з базою даних (на прикладі сутності "User" або "Product") Створення контролера (@RestController) з методами GET/POST/PUT/DELETE; Робота з H2 або PostgreSQL, JPA-репозиторіями; Демонстрація результатів у Postman або Swagger UI.
5	Захист REST API з використанням Spring Security Додавання Spring Security до проєкту; Реалізація простого користувача з ролями в пам'яті; Обмеження доступу до методів контролера за ролями.
6	Юніт- та інтеграційне тестування серверного застосунку, замір часу відповіді Створення тестів за допомогою JUnit + MockMVC; Симуляція повільного методу та вимірювання тривалості; Обговорення продуктивності: де і як виникають затримки.
7	Кешування відповідей REST API, розділення логіки на два сервіси Додавання @Cacheable до сервісного методу; Запуск 2 Spring Boot застосунків, один з яких викликає інший; Пояснення взаємодії між мікросервісами (локально).
8	Використання RestTemplate/WebClient для взаємодії + базовий CI/CD сценарій Використання RestTemplate або WebClient для запитів між сервісами; Ознайомлення з GitLab CI/CD або симуляція збірки з GitHub Actions; Огляд Dockerfile для збірки контейнера.
9	Організація логування та оцінка масштабованості застосунку Підключення логування з Logback або SLF4J; Імітація навантаження та запуск кількох інстансів вручну (на різних портах); Обговорення вертикального/горизонтального масштабування.

6. Самостійна робота студента

1	Розділ 1. Основні поняття серверного програмного забезпечення Тема 1. Вступ до серверного програмного забезпечення. Клієнт-серверна архітектура та роль серверного ПЗ. [5, с. 1-10]. Тема 2. Основи Java для серверної розробки. Огляд синтаксису Java та основ ООП. [6, с. 15-30]. Тема 3. Шаблони проєктування для серверних додатків. Шаблони проєктування (Singleton, Factory). [1, с. 1-50]. Тема 4. Моделі обробки запитів на стороні сервера: синхронна та асинхронна обробка. Ознайомлення з Spring WebFlux, Netty та Vert.x. [7, с. 101-120]. Тема 5. Основи створення веб-сервера (без фреймворків). Переваги використання фреймворків в розробці веб-серверів. [6, с. 220-250].
2	Розділ 2. Основи розробки серверного програмного забезпечення на Java Тема 6. Основи фреймворку Spring. IoC та Dependency Injection. [7, с. 55-80]. Тема 7. Розробка RESTful API з використанням Spring. Основи RESTful сервісів. [7, с. 135-160]. Тема 8. Підключення до баз даних через Spring Data JPA. Огляд Java Persistence API (JPA). [9, с. 1-30]. Тема 9. Основи безпеки серверних додатків. Аутентифікація та авторизація. [4, с. 50-70]. Тема 10. Spring Security: концепції та реалізація. Використання Spring Security для захисту API. [3, с. 150-180]. Тема 11. Тестування серверних додатків. Основи юніт-тестування. [12, с. 1-20]. Тема 12. Обмеження продуктивності: теоретичні та практичні аспекти. Виявлення обмежень без інструментів. [13, с. 200-220].
3	Розділ 3. Прикладні аспекти розробки серверного програмування Тема 13. Кешування в серверних додатках. Використання кешування. [4, с. 100-120]. Тема 14. Основи мікросервісної архітектури. Принципи побудови мікросервісів. [8, с. 10-40]. Тема 15. Взаємодія між мікросервісами. Комунікація через REST. [8, с. 150-180]. Тема 16. Розгортання серверного додатку та CI/CD. Налаштування CI/CD. [6, с. 5-30]. Тема 17. Моніторинг та логування серверних додатків. Інструменти моніторингу. [6, с. 350-380]. Тема 18. Практичні аспекти масштабування серверних додатків. Принципи масштабування. [14, с. 300-320].

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Для успішного проходження курсу та складання контрольних заходів необхідним є вивчення навчального матеріалу за кожною темою. Специфіка курсу передбачає акцент на розумінні підходів і принципів, отримання практичних навичок, а не просто запам'ятовування визначень. Кожен студент повинен ознайомитися і слідувати Положенню про академічну доброчесність, Статуту і розпорядку дня університету. Для успішного засвоєння програмного матеріалу студент зобов'язаний:

- не запізнюватися на заняття;
- не пропускати заняття, а в разі пропуску відновити за допомогою консультування з викладачем та з використанням Кампус конспект, самостійно вивчити матеріал пропущеного заняття та скласти відповідні контрольні заходи в індивідуальному порядку;
- конструктивно підтримувати зворотній зв'язок на всіх заняттях;
- брати активну участь у освітньому процесі;
- своєчасно і старанно виконувати завдання для самостійної роботи;
- бути доброзичливим до однокурсників та викладачів;
- брати участь у контрольних заходах;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в дистанційній on-line формі за погодженням із деканом факультету);
- будь-яке копіювання або відтворення результатів чужої праці (у тому числі списування), якщо тільки робота не має груповий формат, використання чужих завантажених з Інтернету матеріалів кваліфікується як порушення норм і правил академічної доброчесності та передбачає притягнення винного до відповідальності, у порядку, визначеному чинним законодавством та Положенням про академічну доброчесність університету. Результатом невиконання та/або недотримання правил може бути оцінка «не зараховано» за курс.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Максимальна кількість балів з кредитного модуля дорівнює 100.

Рейтинг студента з кредитного модуля складається з балів, отриманих за:

- 1) роботу на лекційних заняттях;
- 2) виконання дев'яти завдань на практичних заняттях;
- 3) виконання модульної контрольної роботи.

1.Робота на лекційних заняттях

На лекційних заняттях може бути проведено бліцопитування студентів щодо виконання домашнього завдання. Такі опитування проводяться на всіх лекційних заняттях, 17 разів протягом семестру, на початку заняття. Ваговий бал за вірну відповідь – 1. Максимальна кількість балів, що може отримати кожен студент за семестр – 17 балів.

2.Виконання завдань на практичних заняттях

Оцінюються 9 завдань, передбачених робочою програмою, за роботу на кожній з яких передбачено максимум 7 балів.

Рейтингові бали кожної роботи складаються з балів за правильність виконання (від 0 до 4) та балів за оформлення звіту роботи (від 0 до 3). За несвоєчасну здачу звіту з виконання завдань на практичних заняттях – штраф 3 бали.

3. Модульна контрольна робота

Максимальний ваговий бал – 20 балів.

Сума вагових балів контрольних заходів протягом семестру складає:

$$R=8+63+20=100 \text{ балів}$$

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Необхідною умовою допуску до заліку є зарахування усіх завдань з практичних занять, а також стартовий рейтинг (rc) не менше 40 % від **R**, тобто 40 балів.

Студенти, які набрали протягом семестру рейтинг з кредитного модуля менше 0,6R, зобов'язані виконувати залікову контрольну роботу.

Студенти, які набрали протягом семестру необхідну кількість балів ($RD \geq 0,6R$), мають можливості:

- отримати залікову оцінку (залік) так званим «автоматом» відповідно до набраного рейтингу (таблиця);
- виконувати залікову контрольну роботу з метою підвищення оцінки (у разі отримання оцінки, більшої ніж «автомат» з рейтингу, студент отримує оцінку за результатами залікової роботи). Оцінювання залікової роботи здійснюється за 100-бальною шкалою

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Що таке серверне програмне забезпечення, і які його ключові характеристики?
2. Які фреймворки використовуються для розробки серверного ПЗ на Java? Наведіть приклади.
3. Які основні класи та API Java використовуються в серверній розробці?
4. Що таке шаблон проєктування Singleton і як його реалізувати на Java?
5. Як використовуються шаблони Factory і Builder у розробці серверних додатків?
6. Які є основні методи створення потоків у Java? Наведіть приклади використання Thread та Runnable.

7. Що таке синхронізація потоків і як вона досягається в Java?
8. Що таке Executor Framework і як він використовується для управління потоками?
9. Як уникати блокування та забезпечувати безпеку потоків у багатопотокових додатках?
10. Що таке Inversion of Control (IoC) і як він реалізований у Spring?
11. Які підходи до конфігурації додатків підтримує Spring? Поясніть відмінності між Java-based, XML та Annotations.
12. Які принципи RESTful сервісів та чому вони важливі для веб-розробки?
13. Як створюються контролери та ресурси у Spring для RESTful API?
14. Які методи HTTP використовуються для взаємодії з REST API і в чому їх різниця?
15. Що таке Java Persistence API (JPA) і як він пов'язаний із Spring Data JPA?
16. Як працює Spring Data JPA для доступу до бази даних?
17. Які основні принципи безпеки для серверних додатків?
18. Які загрози можуть загрожувати серверним додаткам, і як їх уникати?
19. Як Spring Security забезпечує захист REST API?
20. Що таке OAuth2 і які його основні складові?
21. Які інструменти використовуються для тестування REST API з MockMVC?
22. Як здійснюється інтеграційне тестування у Spring-додатках?
23. Як працює збір сміття (Garbage Collection) у Java?
24. Які інструменти використовуються для профілювання продуктивності серверних додатків?
25. Що таке кешування і чому воно важливе для серверних додатків?
26. Як використовувати Spring Cache для кешування даних?
27. Які переваги мікросервісної архітектури та як її реалізувати на Java?
28. Які виклики можуть виникати під час розробки мікросервісів?
29. Як мікросервіси взаємодіють один з одним через REST?
30. Що таке API Gateway і яку роль він відіграє у мікросервісах?
31. Як налаштувати CI/CD з Jenkins та GitLab для серверних додатків?
32. Які інструменти використовуються для моніторингу серверних додатків?
33. Як організувати логування в Spring-додатках?
34. Як аналізувати логи для виявлення проблем у серверних додатках?
35. Які принципи масштабування серверних додатків?
36. У чому різниця між горизонтальним та вертикальним масштабуванням?
37. Які хмарні платформи можна використовувати для масштабування серверних додатків?

Робочу програму навчальної дисципліни (силабус) «Розробка серверного програмного забезпечення»:

Складено професором кафедри інженерії програмного забезпечення в енергетиці НН ІАТЕ, д.т.н., проф. Барабашом Олегом Володимировичем.

Ухвалено кафедрою інженерії програмного забезпечення в енергетиці НН ІАТЕ (протокол № 42 від 18.06.2025 р.)

Погоджено Методичною комісією НН ІАТЕ КПІ імені Ігоря Сікорського (протокол №9 від 27.06.2025 р.)